IN THE SPECIFICATION

Please amend the specification as indicated below by substituting replacement paragraphs that are marked up to show changed made relative to the immediate prior version.

1. At page 10, line 26, please substitute the paragraph beginning with "Figure 1 illustrates circuitry. . ." with the following replacement paragraph:

--Figure 1 illustrates circuitry of an exemplary electronic system or computer system 100. Exemplary computer system 100 includes an address/data bus 99 for communicating information, a central processor 101 coupled with the bus 99 for processing information and instructions, a volatile memory 102 (e.g., random access memory (RAM), static RAM dynamic RAM, etc.) coupled with the bus 99 for storing information and instructions for the central processor 101, and a non-volatile memory 103 (e.g., read only memory (ROM), programmable ROM, flash memory, EPROM, EEPROM, etc.) coupled to the bus 99 for storing static information and instructions for the processor 101. An optional data storage device 104 is also coupled to the bus 99.--

2. At page 14, line 2, please substitute the paragraph beginning with "Additionally, a chip select (CS#) . . ." with the following replacement paragraph:

--Additionally, a chip select (CS#) signal 240 enables the small EEPROM when the signal is low to initiate the transfer of data and marks the end of operation when the signal is high. The controller 105 transfers the CS# signal to the CS# pin [[426]] 427 of Figure 4 to initiate a transfer (CS=0) and to terminate a transfer (CS=1). The controller may continue to read successive bytes by continuing to issue groups of eight serial clock pulses while holding CS=0. An address counter in the small EEPROM automatically increments the address for each eight serial clock pulses.--

3. At page 16, line 9, please substitute the paragraph beginning with "Additionally, a chip select (CS#) . . ." with the following replacement paragraph:

--Additionally, a chip select (CS#) signal 340 enables the large EEPROM when the signal is low to initiate the transfer of data and marks the end of operation when the CS# signal is high in the present embodiment. The controller 105 transfers the CS# signal waveform 340 to the CS# pin [[426]] 427 to initiate a transfer (CS=0) and to terminate a transfer (CS=1). The controller 105 may continue to read successive bytes by continuing to issue groups of eight serial clock pulses while holding CS=0. An address counter in the large EEPROM automatically increments the address for each eight serial clock pulses.--

4. At page 21, line 11, please substitute the paragraph beginning with "Additionally, a chip select (CS#) . . ." with the following replacement paragraph:

--Additionally, a chip select (CS#) signal waveform 510 enables the SPI EEPROM 420 when the signal is low to initiate the transfer of data and marks the end of operation when the CS# signal is high, in the present embodiment of the invention. The controller 105 transfers the CS# waveform 510 to the CS# pin [[426]] 427 to initiate a transfer (CS=0) and to terminate a transfer (CS=1). The controller 105 may continue to read successive bytes by continuing to issue groups of eight serial clock pulses while holding CS=0. An address counter in the SPI EEPROM 420 automatically increments the address for each eight serial clock pulses.--

5. At page 26, line 18, please substitute the paragraph beginning with "However, since the SPI EEPROM is absent . . ." with the following replacement paragraph:

--However, since the SPI EEPROM is absent, the controller also reads a zero byte during the fourth byte (4) 640 as shown in data [1] 645. A zero byte does not correspond to one of the pre-defined non-zero "tag" bytes and the controller 105 has the intelligence to understand that the SPI EEPROM or SPI device is absent. Thus, in the case of a EZ-USB chip, the controller 105 instructs the EZ-USB chip to enumerate using internal values as

in the previous example. Byte (5) 650 and byte (6) 660 would also be zero since the SPI device is absent.--

6. At page 27, line 7, please enter the following new paragraph before the paragraph beginning with "At byte (3) 730 . . . ."

--The SPI controller 105 issues the command, a READ command, in the first byte (1) 710 in the present embodiment. The SPI controller 105 then issues the first 8 bits of the address (00000000) in the second byte (2) 720.--

7. At page 30, line 15, please substitute the paragraph beginning with "Process 800 follows the . . ." with the following replacement paragraph:

Process 800 follows the three stages to read information from the SPI device as discussed previously. These three stages are the command stage, the address stage, and the data stage. Process [[810]] 800 begins with step 810, the command stage, where a memory controller sends the READ command byte in the first byte of data over the bi-directional bus.